Кулешов

1.1

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

1.2

[2]:
```
dataset = pd.read_csv('/content/blood.csv')
dataset.head()
```

[2]:
```
   Recency (months)  Frequency (times)  Monetary (c.c. blood)  Time (months)  \
0                 2                 50                  12500             98
1                 0                 13                   3250             28
2                 1                 16                   4000             35
3                 2                 20                   5000             45
4                 1                 24                   6000             77

   whether he/she donated blood in March 2007
0                                           1
1                                           1
2                                           1
3                                           1
4                                           0
```

[3]:
```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 4].values
print ("          "); print(X[:5])
print ("            "); print(y[:5])
```

```
[[    2    50 12500    98]
 [    0    13  3250    28]
 [    1    16  4000    35]
 [    2    20  5000    45]
 [    1    24  6000    77]]
```

```
[1 1 1 1 0]
```

1.3

```
[4]: # from sklearn.preprocessing import Imputer
# imputer = Imputer(missing_values = 'NaN', strategy = 'mean', axis = 0)
# imputer = imputer.fit(X[:, 1:3])
# X[:, 1:3] = imputer.transform(X[:, 1:3])
# print(X)
```

1.4

1.4.1 (LabelEncoder)

```
[5]: # from sklearn.preprocessing import LabelEncoder
# labelencoder_y = LabelEncoder()
# print("                    ")
# print(y)
# y = labelencoder_y.fit_transform(y)
# print("                      ")
# print(y)
```

1.4.2 OneHotEncoder

```
[ ]: from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder = LabelEncoder()
X[:, 3] = labelencoder.fit_transform(X[:, 3])
onehotencoder = OneHotEncoder(categorical_features = [3])
X = onehotencoder.fit_transform(X).toarray()
print("                     ")
print(X[:4,:])
```

```
[[0.0000000e+00 0.0000000e+00 1.0000000e+00 1.6534920e+05 1.3689780e+05
  4.7178410e+05]
 [1.0000000e+00 0.0000000e+00 0.0000000e+00 1.6259770e+05 1.5137759e+05
  4.4389853e+05]
 [0.0000000e+00 1.0000000e+00 0.0000000e+00 1.5344151e+05 1.0114555e+05
  4.0793454e+05]
 [0.0000000e+00 0.0000000e+00 1.0000000e+00 1.4437241e+05 1.1867185e+05
  3.8319962e+05]]
```

```
D:\ml\Anaconda3\lib\site-packages\sklearn\preprocessing\_encoders.py:363:
FutureWarning: The handling of integer data will change in version 0.22.
Currently, the categories are determined based on the range [0, max(values)],
while in the future they will be determined based on the unique values.
If you want the future behaviour and silence this warning, you can specify
```

```
"categories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the
categories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
D:\ml\Anaconda3\lib\site-packages\sklearn\preprocessing\_encoders.py:385:
DeprecationWarning: The 'categorical_features' keyword is deprecated in version
0.20 and will be removed in 0.22. You can use the ColumnTransformer instead.
  "use the ColumnTransformer instead.", DeprecationWarning)
```

1.5 ,

[6]:
```python
X = X[:, 1:]
print(X[:4,:])
```

```
[[   50 12500    98]
 [   13  3250    28]
 [   16  4000    35]
 [   20  5000    45]]
```

1.6

[7]:
```python
# from sklearn.cross_validation import train_test_split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

1.7

[8]:
```python
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

[8]:
```
LinearRegression()
```

1.7.1 ,

[9]:
```python
y_pred = regressor.predict(X_test)
print(y_pred)
```

```
[ 0.19582029  0.34874368  0.34497978  0.38805485  0.87983108  0.2916319
  0.21280578  0.2916319   0.13418344  0.41216059  0.21143328  0.17191833
  0.27444263  0.41394065  0.342031    0.12253813  0.22999506  0.17685096
  0.07966684  0.27149385  0.12075807  0.25567708  0.228215    0.26416982
  0.2060931   0.3434035   0.27444263  0.42462102  0.19404023  0.13103088
  0.3914112   0.152949    0.08953209  0.2422517  -0.0558635   0.26416982
  0.20253298  0.19999175  0.22999506  0.20253298  0.25567708  0.19938041
```

```
 0.26416982  0.47044109  0.06919026  0.01446987  0.18575126  0.24026786
 0.46194835  0.26635744  0.2060931   0.28176665  0.24026786  0.228215
 0.18910761  0.28176665  0.07590294  0.30210849  0.13774357  0.24026786
 0.09665234  0.27444263  0.27820653  0.22725005  0.26416982  0.26416982
 0.24026786  0.24026786 -0.13468962  0.58090075  0.37640954  0.134591
 0.27444263  0.19088767  0.10158496  0.26416982  0.32265411  0.22999506
 0.34045472  0.28847934  0.34497978  0.31833282  0.26101726  0.0725466
 0.3123813   0.32306166  0.26416982  0.27820653  0.19582029 -0.09537845
 0.22328237  0.01090975  0.59096978  0.32107782  0.99639208  0.35388009
 0.42578974 -0.00251562  0.3951751   0.27444263  0.3123813   0.3509313
 0.19404023  0.25567708  0.21280578  0.06405385  0.16657815  0.25567708
 0.285123    0.16520565  0.27444263  0.17685096  0.16006924  0.21280578
 0.3641529   0.08637953  0.28491922  0.30190471  0.17685096 -0.11750034
 0.21143328  0.23731908  0.26772994  0.13596351  0.33984338  0.20431304
 0.30190471  0.26416982  0.30114355  0.30566861  0.34360728  0.25745714
 0.28491922  0.29341196  0.209857    0.30902496  0.09309221  0.2060931
 0.16835821  0.2384878   0.29046318  0.23355518  0.28491922  0.30922873
 0.1567129   0.26416982  0.23355518  0.15986546  0.21478962  0.20253298]
```

[ ]:

```python
import statsmodels.formula.api as sm
X = np.append(arr = np.ones((50, 1)).astype(int), values = X, axis = 1)
X_opt = X[:, [0, 1, 2, 3, 4, 5]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()
```

[ ]:

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.951
Model:                            OLS   Adj. R-squared:                  0.945
Method:                 Least Squares   F-statistic:                     169.9
Date:                Thu, 01 Sep 2022   Prob (F-statistic):           1.34e-27
Time:                        17:45:14   Log-Likelihood:                -525.38
No. Observations:                  50   AIC:                             1063.
Df Residuals:                      44   BIC:                             1074.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       5.013e+04   6884.820      7.281      0.000    3.62e+04     6.4e+04
x1           198.7888   3371.007      0.059      0.953   -6595.030    6992.607
x2           -41.8870   3256.039     -0.013      0.990   -6604.003    6520.229
x3             0.8060      0.046     17.369      0.000       0.712       0.900
x4            -0.0270      0.052     -0.517      0.608      -0.132       0.078
```

```
x5             0.0270      0.017      1.574      0.123      -0.008       0.062
==============================================================================
Omnibus:                       14.782   Durbin-Watson:                   1.283
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.266
Skew:                          -0.948   Prob(JB):                     2.41e-05
Kurtosis:                       5.572   Cond. No.                     1.45e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.45e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

[ ]:

```python
X_opt = X[:, [0, 1, 3, 4, 5]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()
```

[ ]:

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.951
Model:                            OLS   Adj. R-squared:                  0.946
Method:                 Least Squares   F-statistic:                     217.2
Date:                Thu, 01 Sep 2022   Prob (F-statistic):           8.49e-29
Time:                        17:45:18   Log-Likelihood:                -525.38
No. Observations:                  50   AIC:                             1061.
Df Residuals:                      45   BIC:                             1070.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       5.011e+04   6647.870      7.537      0.000    3.67e+04    6.35e+04
x1           220.1585   2900.536      0.076      0.940   -5621.821    6062.138
x2             0.8060      0.046     17.606      0.000       0.714       0.898
x3            -0.0270      0.052     -0.523      0.604      -0.131       0.077
x4             0.0270      0.017      1.592      0.118      -0.007       0.061
==============================================================================
Omnibus:                       14.758   Durbin-Watson:                   1.282
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.172
Skew:                          -0.948   Prob(JB):                     2.53e-05
Kurtosis:                       5.563   Cond. No.                     1.40e+06
==============================================================================
```

```
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.4e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

[ ]:
```
X_opt = X[:, [0, 3, 4, 5]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()
```

[ ]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.951
Model:                            OLS   Adj. R-squared:                  0.948
Method:                 Least Squares   F-statistic:                     296.0
Date:                Thu, 01 Sep 2022   Prob (F-statistic):           4.53e-30
Time:                        17:45:23   Log-Likelihood:                -525.39
No. Observations:                  50   AIC:                             1059.
Df Residuals:                      46   BIC:                             1066.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       5.012e+04   6572.353      7.626      0.000    3.69e+04    6.34e+04
x1             0.8057      0.045     17.846      0.000       0.715       0.897
x2            -0.0268      0.051     -0.526      0.602      -0.130       0.076
x3             0.0272      0.016      1.655      0.105      -0.006       0.060
==============================================================================
Omnibus:                       14.838   Durbin-Watson:                   1.282
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.442
Skew:                          -0.949   Prob(JB):                     2.21e-05
Kurtosis:                       5.586   Cond. No.                     1.40e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.4e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

[ ]:
```
X_opt = X[:, [0, 3, 5]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
```

```
regressor_OLS.summary()
```

[ ]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.950
Model:                            OLS   Adj. R-squared:                  0.948
Method:                 Least Squares   F-statistic:                     450.8
Date:                Thu, 01 Sep 2022   Prob (F-statistic):           2.16e-31
Time:                        17:45:28   Log-Likelihood:                -525.54
No. Observations:                  50   AIC:                             1057.
Df Residuals:                      47   BIC:                             1063.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       4.698e+04   2689.933     17.464      0.000    4.16e+04    5.24e+04
x1             0.7966      0.041     19.266      0.000       0.713       0.880
x2             0.0299      0.016      1.927      0.060      -0.001       0.061
==============================================================================
Omnibus:                       14.677   Durbin-Watson:                   1.257
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.161
Skew:                          -0.939   Prob(JB):                     2.54e-05
Kurtosis:                       5.575   Cond. No.                     5.32e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.32e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

[ ]:
```
X_opt = X[:, [0, 3]]
regressor_OLS = sm.OLS(endog = y, exog = X_opt).fit()
regressor_OLS.summary()
```

[ ]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.947
Model:                            OLS   Adj. R-squared:                  0.945
Method:                 Least Squares   F-statistic:                     849.8
Date:                Thu, 01 Sep 2022   Prob (F-statistic):           3.50e-32
```

```
Time:                        17:45:32   Log-Likelihood:                -527.44
No. Observations:                  50   AIC:                             1059.
Df Residuals:                      48   BIC:                             1063.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       4.903e+04   2537.897     19.320      0.000    4.39e+04    5.41e+04
x1             0.8543      0.029     29.151      0.000       0.795       0.913
==============================================================================
Omnibus:                       13.727   Durbin-Watson:                   1.116
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               18.536
Skew:                          -0.911   Prob(JB):                     9.44e-05
Kurtosis:                       5.361   Cond. No.                     1.65e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.65e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```